

Agentic RAG Chatbot for Multi-Format Document QA using MCP

Technologies: Python, Streamlit, FAISS, Ollama, MCP



Problem Statement & Objective

Goal: Build an intelligent chatbot that answers user questions using uploaded documents (PDF, DOCX, CSV, PPTX, TXT)

Key Features:

- Agent-based modular architecture
- Uses Retrieval-Augmented Generation (RAG)
- Agents communicate via Model Context Protocol (MCP)
- Supports multi-turn chat with contextual awareness

Agent-Based Architecture

Agents Involved:

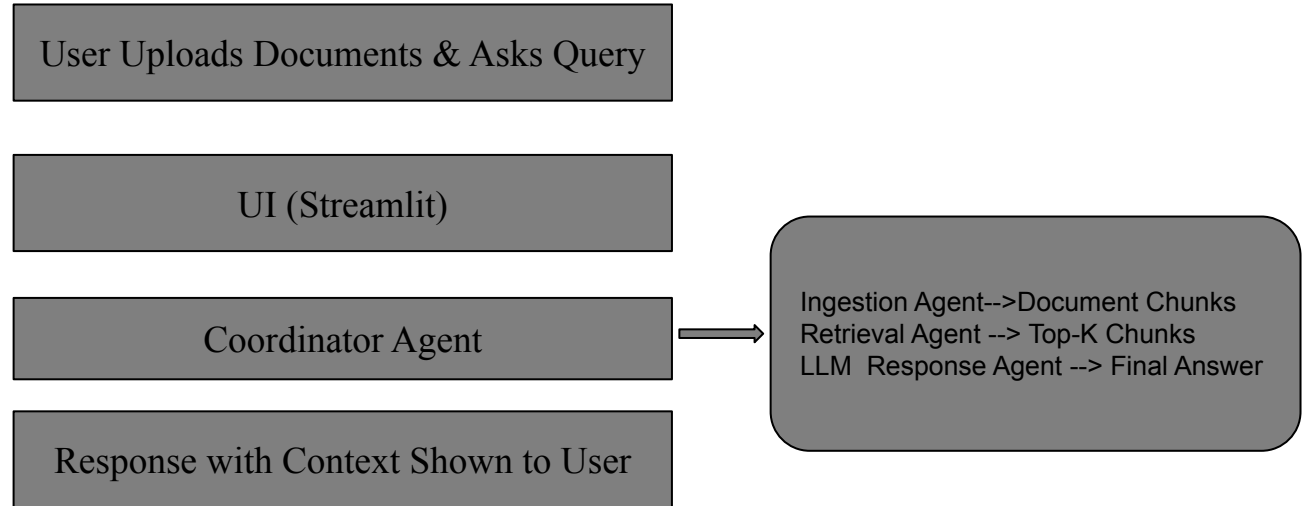
1. **IngestionAgent:** Parses and preprocesses documents
2. **RetrievalAgent:** Embeds and retrieves relevant context
3. **LLMResponseAgent:** Forms prompt, queries LLM, returns response

CoordinatorAgent: Manages message flow between agents using MCP

MCP Message Format:

```
{
  "sender": "RetrievalAgent",
  "receiver": "LLMResponseAgent",
  "type": "RETRIEVAL_RESULT",
  "trace_id": "rag-457",
  "payload": {
    "retrieved_context": ["slide 3...", "doc summary..."],
    "query": "What KPIs were tracked in Q1?"
  }
}
```

System Workflow Diagram



Challenges & Future Scope

Challenges Faced:

- Parsing formatting-heavy files (PPTX, PDF)
- Chunking long documents intelligently
- Prompt engineering to avoid LLM hallucinations

Future Scope:

- Add OCR for scanned PDFs
- Migrate to LangChain + LangGraph
- Use message queue or REST for MCP
- Deploy with Docker + Kubernetes

📄 Multi-Format Agentic RAG Chatbot

Upload your documents



Drag and drop files here

Limit 200MB per file • PDF, DOCX, PPTX, CSV, TXT, MD

[Browse files](#)



Tithi_Resume.pdf 232.1KB



Ask a question

Submit

